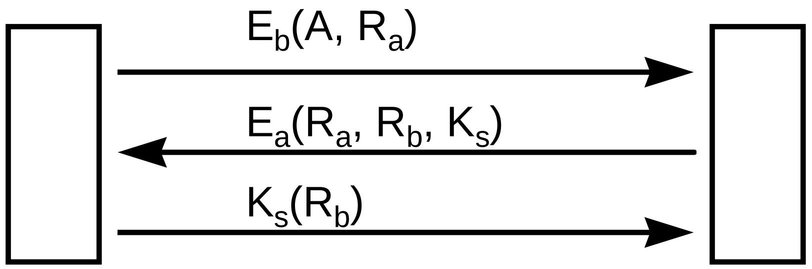
# Announcements

- Reading
  - Today: 7.2, 7.3 (skip 7.3.2 and 7.3.4)
  - Tuesday: 7.4, 7.5
- Will have office hours on Thursday 10:45 to 11:45
- Problems for Chapter 4:
  - 4, 10, 17, 18, 21, 37, 40

# Authentication using Public Keys

- Assume each party knows the other's public key



- How To learn others Public Key?
  - use a public key server
    - but how do we trust the public key server?
    - have a public key for the public key server
    - possible to have man-in-the-middle attacks
  - interlock protocol
    - only send half the message (odd bits) at a time
    - prevents man-in-the-middle attacks
    - still possible to spoof service

# Digital Signatures

- Want to “sign” a message such that:
  - receiver can verify the identity of the sender
  - sender cannot repudiate the contents of the message
  - receiver cannot forge a message
- Central authority (BB)
  - A sends BB A, $K_a(B, R_a, t, P)$
  - BB sends B $K_b(A, R_a, t, P, K_{bb}(A, t, P))$
  - everyone trusts BB
    - BB can be called on to decrypt messages to verify them
    - BB need not store all message that it validates
  - t - timestamp used to prevent replay attacks
- Public Key
  - need E(D(P)) = P **and** D(E(P)) = P
  - A sends B $E_b(D_a(P))$
    - B keeps $D_a(P)$ and thirdy party can use $E_a$ to verify it’s from A

# Digital Signatures (cont.)

- Problems
  - Repudiation
    - inform police that the key was stolen
    - claim the “bad guy” sent the message
  - Key Changes
    - need to keep records of when keys were in use
- Standards
  - RSA Algorithm
    - popular with many commercial systems
  - El Gamal
    - NSA/NIST Standard
    - too new, and private to have trust

# Message Digests

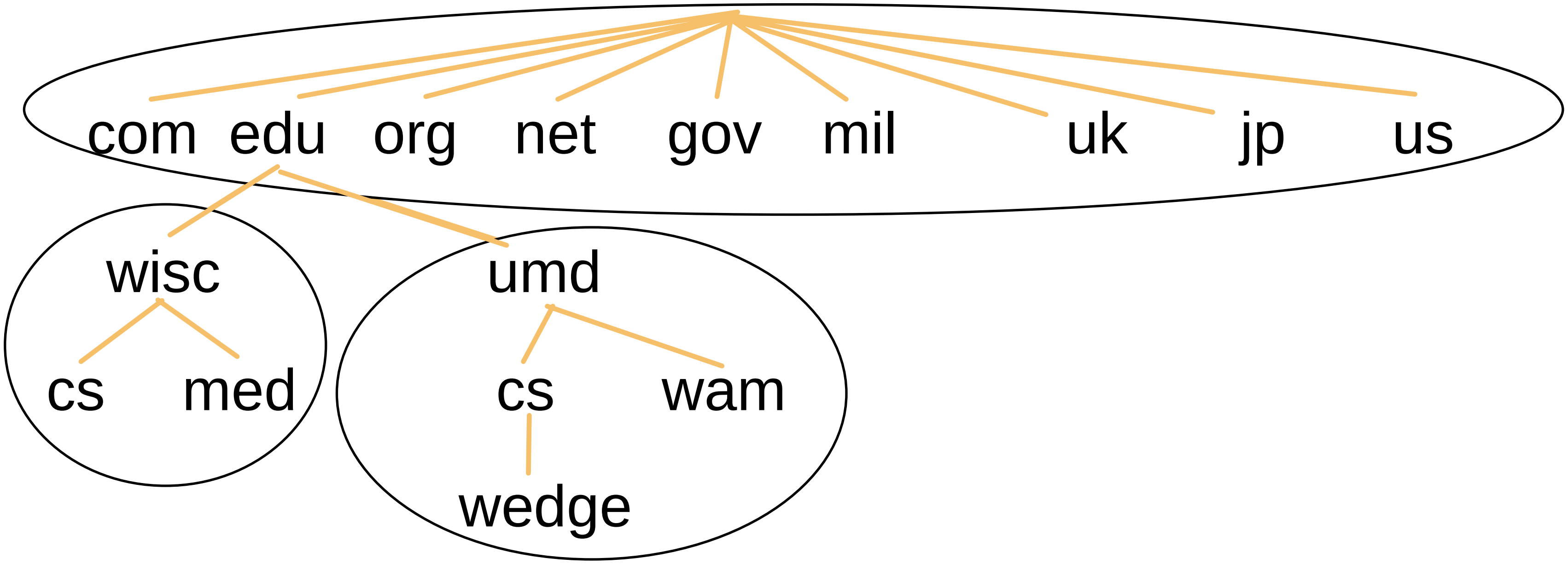
- Goal: Send Signed Plain text
  - can use slow cryptography on signature since its short
- Need:
  - Given P, easy to compute MD(P)
  - Given MD(P), impossible to find P
  - no P and P’ exist such that MD(P) = MD(P’)
    - use hash functions that produce >= 128 bit digest
- Operation
  - A sends P, $D_a$(MD(P))
- Digest Functions
  - MD5
    - produces 128 bit digest
  - SHS
    - NSA/NIST effort
    - produces 160 bit output

# Naming Hosts In the Internet

- Originally used a single file
  - all hosts had line line with name and IP Address
- Domain Naming System (DNS)
  - introduced in 1986
  - tree based structure to names
  - Names
    - full name must be less than 256 characters
    - each part can be up to 64 characters
    - are case insensitive
  - administration of subtrees can be deligated
    - each administrative region is called a zone

# Examples of Domain Names

- Domains can be both roots of subtrees **and** hosts
  - – For example: cs.umd.edu
- Top level country codes
  - – required by PTTs outside of US



# DNS (cont.)

- Resource Records
  - DNS is really a distributed, replicated database
- Several types of tuples in the database
  - SOA - Start of Authority information for a zone
  - A - IP Address record
  - MX - Mail exchanger
    - priority and destination (host name)  to accept mail
  - NS - Name of the name server for this domain
  - CNAME - Canonical name (DNS name)
  - PTR - alias for an IP Address
  - HINFO - Host Info (CPU and OS type information)
  - TXT - other text information

# Name Servers

- A collection of servers is used to run DNS
  - root servers: handle top level domains
  - have pointers to servers for deligated sub-domains
  - areas of the namespace covered by a server called a zone
- Zones
  - has one primary server (zone information stored on disk)
  - secondary name servers (get info from primary)
    - secondary server may be located outside of the zone
- Namelookup
  - start at current name server
  - if not found, resolve down tree to correct zone server
  - data may be cached in servers
    - this information may be out of data
    - **authoritative data** comes from the primary/secondary NS